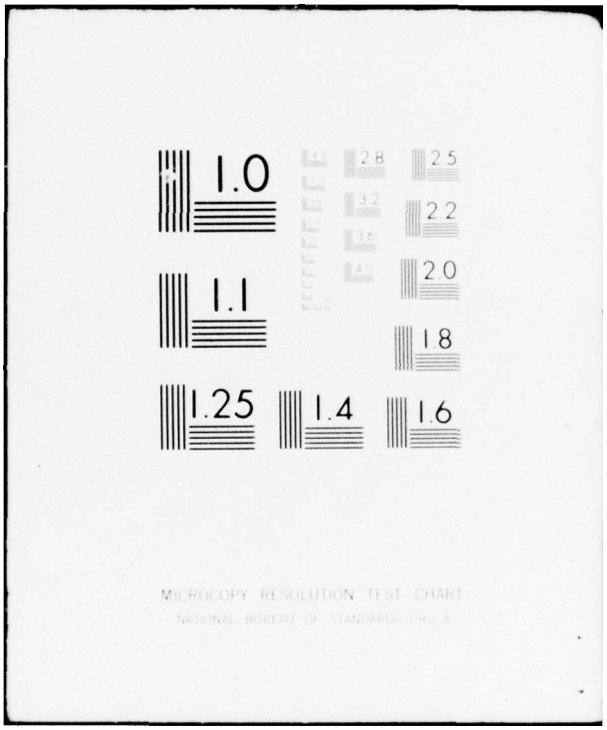
AD-A046 457 COLORADO STATE UNIV FORT COLLINS DEPT OF MATHEMATICS F/G 12/1
OPTIMAL PARTITIONING OF NEWTON'S METHOD FOR CALCULATING ROOTS.(U)
MAY 77 G MEINARDUS, G D TAYLOR AFOSR-76-2878

UNCLASSIFIED AFOSR-TR-77-1261 NL

| OF |
AD
A046457

END
DATE
FILMED
12-77
DDC


1.0
1.1
1.25
1.4
1.6
2.8
2.5
3.2
2.2
3.6
2.0
4.0
1.8
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

OPTIMAL PARTITIONING OF NEWTON'S METHOD FOR CALCULATING ROOTS

Günter Meinardus

AND

G. D. Taylor[1]

Department of Mathematics, Colorado State University, Ft. Collins, Colorado, U.S.A.

SUMMARY

In this paper an algorithm for calculating roots is given that is Newton's method initialized with a piecewise best starting approximation. The piecewise best starting approximation corresponds to a partition of the interval of the domain of Newton's method and it is shown how to choose this partition to be optimal. Explicit formulas are given when linear polynomials are used for the best starting approximations. Specific examples are given for square roots, cube roots and reciprocal square roots.

1. INTRODUCTION

An effective algorithm for calculating roots is Newton's method initialized with a best starting approximation. Recently [2,3], this procedure has been modified in that a piecewise best starting approximation was used for initializing the Newton iteration. This is equivalent to subdividing the interval of application of the Newton iteration into subintervals and applying the theory of best starting approximations to each subinterval. In this paper, we shall describe how this subdivision can be done in an optimal manner.

---

[1] Research sponsored by the Air Force Office of Scientific Research, Air Force Systems Command, USAF, under Grant No. AFOSR-76-2878.

The theory of best starting approximations for calculating roots was first studied by Moursund [10] for the special case of square roots. This theory was extended to general roots by Moursund and Taylor in [11]. Subsequent studies found that the best starting approximation for calculating roots via Newton's method is independent of the number of iterations to be used and is, in fact, a multiple of the best relative approximation to the root [8, 12, 13, 14, 15, 17]. Surprisingly, it was also shown [8, 12, 14] that one of the square root subroutines in use prior to the development of this theory [5], was, in fact, the method of Moursund.

This theory allows considerable leeway in desigining a specialized root routine. In the case of large scale computers, it is possible to design routines that return a predetermined accuracy and in most cases have a decreased time lag when compared to calling the system's subroutine. Whereas, for microprocessors these algorithms can be incorporated as firmware support whenever the ability to calculate roots is desired. For example, in [2,3] square root routines of this type were developed for 8-bit and 16-bit microprocessors, where it was desired for the 16-bit routine to develop an algorithm which would give 15 bits of accuracy after one Newton iteration initialized with a linear polynomial and have as its domain of application all numbers of the form $X = \{\frac{j}{2^{16}}\}_{j=2^{14}+1}^{2^{16}}$. In order to develop such an algorithm it was necessary to partition the point set into $X = X_1 \cup X_2 \cup X_3$, $X_1 = X \cap (1/4, c_1]$, $X_2 = X \cap (c_1, c_2]$ and $X_3 = X \cap (c_2, 1]$ for appropriately chosen c_1 and c_2. Treating each of these three sets independently it is possible to develop a square root algorithm satisfying the constraints listed above (with the exception of the domain constraint). Thus, the algorithm used a piecewise best linear polynomial starting approximation -- defined to be optimal on X_1, X_2 and X_3 separately. The actual use

of this algorithm would first involve a scaling of the given positive number by an even power of 2 to produce a number in X. Next, it would determine which subset (X_1, X_2 or X_3) contains the scaled number and then evaluate the best linear starting approximation at the scaled number. This value is then used to initialize one Newton iteration for calculating square roots. The result of this iteration is then multiplied (shifted) by 2 to half the power of the original scaling and then this value is returned as the desired square root. This algorithm was compared with the corresponding direct and Cordic type of methods [16] including Chen's modified version [4] and was found to be preferable for the types of architectures considered. A second example was in the development of a reciprocal square root routine using a divide-free Newton iteration for inclusion in the particle moving section of a relativistic plasma code on an IBM 360/91. The design constraints in this case were a required accuracy of 10^{-5} after one Newton iteration initialized with a linear polynomial on [1/8, 1/2]. Here the interval [1/8,1/2] was divided into five subintervals in order to satisfy these constraints.

The main result of this paper is the following: Suppose that one wishes to subdivide an interval [a, b] into ν subintervals and calculate a root via Newton's method on [a, b] by calculating independently on each subinterval. Then there exists a unique partition that is optimal with respect to the relative error. Even though this result is for the continuous case it is still useful for actual applications since it suggests where to subidivide in a discrete setting. The precise organization of this paper is as follows. Section 2 contains a summary of the definitions and basic theoretical results of best starting approximations, section 3 gives our general results and section 4 gives specific examples of this theory for

calculating square roots, reciprocal square roots using a divide free iteration and cube roots, each subject to certain design constraints.

2. DEFINITIONS AND BASIC NOTIONS

Let [a,b] be a fixed interval with $0 < a < b$ and set

$$\mathcal{R}_n^m[a,b]=\{R=P/Q : P \in \Pi_m,\ Q \in \Pi_n,\ Q(x) > 0 \text{ for all } x \in [a,b],\ (P,Q)=1\}$$

where Π_k denotes the class of all real algebraic polynomials of degree less than or equal to k and (P,Q) denotes the greatest common (polynomial) divisor of P and Q. Fix α a real number, $\alpha \neq 0$ or ± 1, and define N_α: $C^+[a,b] \to C[a,b]$, where $C^+[a,b]$ denotes the class of all continuous positive functions defined on [a,b], by

$$N_\alpha(h)(x) = \alpha\left[(\beta - 1)h(x) + \frac{x}{h^{\beta-1}(x)}\right],\ \beta = 1/\alpha.$$

Observe that $N_\alpha(h)(x)$, for fixed x is simply the result of one Newton iteration for calculating x^α with h(x) as its starting approximation (or initial guess). That is, the formula for N_α is simply the result of applying Newton's method to $y^\beta - x = 0$, x fixed. As usual, we also define N_α^k by $N_\alpha^k(h)(x) = N_\alpha(N_\alpha^{k-1}(h))(x)$, the result of k Newton iterations. Then $R^* \in \mathcal{R}_n^m[a, b]$ is said to be the best (relative) starting approximation from $\mathcal{R}_n^m[a, b]$ for calculating α-roots on [a, b] provided

$$(1)\quad \eta_\alpha[a,b]= \left\|\frac{x^\alpha - N_\alpha(R^*)(x)}{x^\alpha}\right\|_{[a,b]} = \min_{R \in \mathcal{R}_n^m[a,b]} \left\|\frac{x^\alpha - N_\alpha(R)(x)}{x^\alpha}\right\|_{[a,b]}$$

where $\|f(x)\|_{[a,b]} = \max\{|f(x)| : x \in [a,b]\}$ for $f \in C[a, b]$. We shall suppress the subscript [a, b] on $\|\cdot\|$ whenever the meaning is clear. Thus the relative error of approximating x^α with one Newton iteration for calculating x^α with initial guess R(x) is minimized on the interval [a,b] if $R^*(x)$ is used as the initial guess. It is shown in [6,9] that R^* exists, is unique and is

a multiple (depending upon α, [a, b], m and n) of the best relative approximation, $\tilde{R}(x)$, to x^α from $\mathcal{R}_n^m[a, b]$, i.e.

(2) $$\left\| \frac{x^\alpha - \tilde{R}(x)}{x^\alpha} \right\| = \min_{R \in \mathcal{R}_n^m[a,b]} \left\| \frac{x^\alpha - R(x)}{x^\alpha} \right\| .$$ From the general theory of uniform relative approximation [1], it is known that $\tilde{R}(x)$ exists, is unique and can be calculated by various methods (see for example, [7]). In fact [6,9], if $\left\| \frac{x^\alpha - \tilde{R}(x)}{x^\alpha} \right\| = \lambda_\alpha$ then

$R^*(x) \equiv \gamma_\alpha \tilde{R}(x)$ where

(3) $$\gamma_\alpha = \left[\left((1 + \lambda_\alpha)^{\beta-1} - (1 - \lambda_\alpha)^{\beta-1} \right) / 2(\beta - 1)\lambda_\alpha (1 - \lambda_\alpha^2)^{\beta-1} \right]^\alpha , \quad \beta = 1/\alpha$$

and this same $R^*(x)$ is also the best starting approximation from $\mathcal{R}_n^m[a,b]$ for k Newton iterates, i.e.

$$\eta_\alpha^k[a,b] = \left\| \frac{x^\alpha - N_\alpha^k(R^*)(x)}{x^\alpha} \right\| = \min_{R \in \mathcal{R}_n^m[a,b]} \left\| \frac{x^\alpha - N_\alpha^k(R)(x)}{x^\alpha} \right\| .$$

In closing this section, we would like to remark that a theory of best (absolute) starting approximations for calculating roots, i.e.

$$\inf_{R \in \mathcal{R}_n^m[a,b]} \| x^\alpha - N_\alpha^k(R)(x) \| ,$$

k a positive integer is neither as well developed nor as rich as the corresponding relative theory. It is known [9] that best absolute starting approximations exist, are unique and can (in theory) be calculated by a Remes type algorithm or a generalized differential correction algorithm [7]. Whether or not best absolute starting approximations are a multiple of some other well known approximation to x^α is not known (they are not a multiple of the best uniform approximation to x^α) and optimal partitioning results corresponding to what we shall prove for the relative case are not known. Thus, unless explicitly stated to the contrary, we shall be concerned with the relative theory in what follows.

3. MAIN RESULTS

In this setting, we wish to first prove

Theorem 1. If $R^* \in \mathcal{R}_n^m[a,b]$ is the best starting approximation from $\mathcal{R}_n^m[a,b]$ for calculating α-roots on $[a,b]$ then $\hat{R}(t) = \rho^\alpha R^*(t/\rho)$, $\rho a \le t \le \rho b$, $\rho > 0$ is the best starting approximation from $\mathcal{R}_n^m[\rho a, \rho b]$ for calculating α-roots on $[\rho a, \rho b]$. Furthermore, $\eta_\alpha[a,b] = \eta_\alpha[\rho a, \rho b]$.

Proof: We are given that $\eta_\alpha[a,b] = \left\| \frac{x^\alpha - N_\alpha(R^*)(x)}{x^\alpha} \right\|$.

Define the change of variables $x = t/\rho$. Then, for $\rho a \le t \le \rho b$, by direct substitution we have that

$$\left\| \frac{t^\alpha - N_\alpha(\hat{R})(t)}{t^\alpha} \right\|_{[\rho a,\rho b]} = \left\| \frac{x^\alpha - N_\alpha(R^*)(x)}{x^\alpha} \right\|_{[a,b]}$$

$$= \min_{R \in \mathcal{R}_n^m[a,b]} \left\| \frac{x^\alpha - N_\alpha(R)(x)}{x^\alpha} \right\|_{[a,b]}$$

$$= \min_{\rho^\alpha R \in \mathcal{R}_n^m[\rho a,\rho b]} \left\| \frac{t^\alpha - N_\alpha(\rho^\alpha R)(t/\rho)}{t^\alpha} \right\|_{[\rho a,\rho b]}$$

$$= \eta_\alpha[\rho a,\rho b].$$

Thus, by definition $\hat{R}(t) = \rho^\alpha R^*(t/\rho)$ is the best starting approximation from $\mathcal{R}_n^m[\rho a, \rho b]$ for calculating α-roots on $[\rho a, \rho b]$ and, by the first comment of the proof, $\eta_\alpha[a, b] = \eta_\alpha[\rho a, \rho b]$. ∎

Using this result, we are able to prove our optimal partitioning result. The flavor of this result is the following. Suppose that one wishes to subdivide the interval $[a, b]$, $0 < a < b$, into ν subintervals and calculate α-roots on $[a, b]$ by actually calculating α-roots independently on each subinterval. Then, it turns out, that there exists a unique partitioning of $[a, b]$ into ν subintervals such the relative error of

approximating x^{α} with a Newton iterate initialized on each subinterval with the best starting approximations for that subinterval is minimal over all such partitions of [a, b] and, in fact, the relative error on each subinterval is the same.

Theorem 2. Let $\mathcal{P} = \{\underline{d} : \underline{d} = \{d_0, d_1, \ldots, d_\nu\}$ with $a = d_0 < d_1 < d_2 < \ldots < d_{\nu-1} < d_\nu = b\}$ be the set of all partitions of [a, b] into ν subintervals. Then, there exists one and only one partition, $\underline{c} = \{c_0, c_1, \ldots, c_\nu\} \in \mathcal{P}$ for which

$$\max_{0\le i\le \nu-1} \eta_\alpha[c_i, c_{i+1}] = \min_{\underline{d}\in\mathcal{P}} \max_{0\le i\le \nu-1} \eta_\alpha[d_i, d_{i+1}].$$

This unique partition is given by the formulas $c_j = a^{(\nu-j)/\nu} b^{j/\nu}$, $j=0,1,\ldots,\nu$. In addition, this theorem holds with η_α replaced by η_α^k for k a positive integer and $\eta_\alpha^k[c_i, c_{i+1}] = \eta_\alpha^k[c_{i+1}, c_{i+2}]$ for all k, α and i.

Proof: First, observe that for the partition, $\underline{c} = \{c_j\}_{j=0}^{\nu}$, we have that $[c_\mu, c_{\mu+1}] = [\rho_\mu c_0, \rho_\mu c_1]$ where $\rho_\mu = (\frac{b}{a})^{\mu/\nu}$, for $\mu = 1, \ldots, \nu - 1$. Thus, by Theorem 1, $\eta_\alpha[c_\mu, c_{\mu+1}] = \eta_\alpha[c_0, c_1]$ for $\mu = 1, \ldots, \nu - 1$.

To prove the minmax statement of Theorem 2 (which will also establish the uniqueness claim) we prove the following result first. Namely, if [a, b], $0 < a < b$ and [c, d], $0 < c < d$ are any two intervals and $\frac{a}{b} > \frac{c}{d}$ then $\eta_\alpha[a, b] > \eta_\alpha[c, d]$. Now by Theorem 1, we can replace [c, d] by [ρc, ρd] where $\rho = \frac{a}{c}$ and $\eta_\alpha[c, d] = \eta_\alpha[\rho c, \rho d]$. Thus setting $e=\rho d<b$, we shall prove that $\eta_\alpha[a, b] > \eta_\alpha[a, e]$ which will establish this result. To do this, let $\tilde{R}(x)$ be the best relative approximation to x^α on [a, b] from $\mathcal{R}_n^m[a, b]$. Define the defect, $\tilde{d}$, of $\tilde{R}(x)$ by $\tilde{d} = \min(m - \partial\tilde{P}, n - \partial\tilde{Q})$

where $\tilde{R}(x) = \tilde{P}(x)/\tilde{Q}(x)$ and ∂P denotes the exact degree of the polynomial P. Then, by the standard theory of best relative approximation [1], there exists at least $N = n + m + 2 - \tilde{d}$ extreme points, $a \leq x_1 < x_2 < \ldots < x_N \leq b$ on which the error curve $E(x) = 1 - \frac{\tilde{R}(x)}{x^\alpha}$ alternates, i.e., $|E(x_i)| = \|E\|$, $i = 1, \ldots, N$ and $E(x_i) = -E(x_{i+1})$, $i = 1, 2, \ldots, N - 1$. Since $E \in C^1[a, b]$, we must have that $E'(x_i) = 0$ for at least $i=2,\ldots,N-1$. Thus, $E'(x)$ must have at least $n + m - d$ zeros. Now,

$$E'(x) = \frac{x^{\alpha-1}[x\tilde{Q}(x)\tilde{P}'(x) - \tilde{P}(x)(\tilde{Q}(x) + x\tilde{Q}'(x))]}{(x^\alpha \tilde{Q}(x))^2}$$

and since $a > 0$, and the degree of the polynomial in the brackets in the numerator equals $\partial\tilde{P} + \partial\tilde{Q}$ which is less than or equal to $n + m - 2\tilde{d}$, we see that $E'(x)$ can have at most $n + m - 2\tilde{d}$ zeros in $[a, b]$. Comparing these two zero counts, we see that we must have $\tilde{d} = 0$, $\partial\tilde{P} = m$, $\partial\tilde{Q} = n$, $x_1 = a$, $x_N = b$ and that $E(x)$ must have precisely $N = n + m + 2$ extreme points in $[a, b]$. Here, $y \in [a, b]$ is said to be an extreme point if $|E(y)| = \|E\|$. Thus, $\tilde{R}$ is not the best relative approximation to x^α on $[a, e]$ from $\mathcal{R}_m^n[a, e]$ since $E(x)$ does not have the necessary alternating behavior on $[a, e]$. Let $R_1 \in \mathcal{R}_m^n[a, e]$ be the unique best relative approximation to x^α on $[a, e]$. We claim that R_1 is not a multiple of $\tilde{R}$. This follows from the above zero counting argument, since for any real $c \neq 0$, $E_c(x) = 1 - \frac{c\tilde{R}(x)}{x^\alpha}$ must be such that $E_c'(x)$ vanishes in $[a, b]$ only where $E'(x)$ vanishes in $[a, b]$, implying that $c\tilde{R}$ does not have the necessary number of alternations to be the best relative approximation to x^α in $[a, e]$. Since the best starting approximations for calculating α-roots from $\mathcal{R}_n^m[a, e]$ and $\mathcal{R}_n^m[a, b]$ on $[a, e]$ and $[a, b]$ are multiples of R_1 and $\tilde{R}$, respectively and $\tilde{R} \in \mathcal{R}_m^n[a, e]$ we must have that

$$\eta_\alpha[a, e] = \min_{R \in \mathcal{X}_n^m[a,e]} \left\| \frac{x^\alpha - N_\alpha(R)(x)}{x^\alpha} \right\|_{[a,e]} < \eta_\alpha[a, b]$$

by uniqueness.

Now let $\underline{d} = \{d_j\}_{j=0}^{\nu}$, $a = d_0 < d_1 < \ldots < d_\nu = b$, be a different partition of $[a, b]$ than the one given by the formulas in the hypothesis of Theorem 2. Then we claim that for some j, $0 \le j \le \nu - 1$, $\frac{a}{d_j} d_{j+1} > c_1$ must hold. Indeed, if $\frac{a}{d_j} d_{j+1} \le c_1$ for all j with strict inequality holding at least once, say at j_1, $0 \le j_1 \le \nu - 1$ where j_1 is the first index where strict inequality holds (such an index must exist since the partitions are distinct), then we must have that $d_j \le c_j$ for $0 \le j \le j_1$ and $d_j < c_j$ for $j_1 < j \le \nu$. This will give a contradiction since we must have $d_\nu = c_\nu = b$. Now, this assertion is proved by an inductive argument as follows. Since $d_0 = c_0 = a$, we assume that $d_j \le c_j$ hold for some j, $0 \le j \le j_1 - 1$ then $d_{j+1} \le \frac{c_1}{a} d_j$ by our original assumption so that $d_{j+1} \le c_0^{-1/\nu} c_\nu^{1/\nu} d_j \le c_0^{-1/\nu} c_\nu^{1/\nu} c_j = c_{j+1}$. Thus, for $0 \le j \le j_1$ we have $d_j \le c_j$. For $j = j_1$, the assumption $\frac{a}{d_j} d_{j+1} \le c_i$ implies $d_{j_1+1} < \frac{c_1}{a} d_{j_i}$ $\le c_0^{-1/\nu} c_\nu^{1/\nu} c_{j_1} = c_{j_1+1}$ and then, by induction again, we find that $d_j < c_j$ for $j_1 < j \le \nu$ as claimed. Thus, $\frac{a}{d_j} d_{j+1} \le c_1$ cannot hold for all j.

Let j, $0 \le j \le \nu - 1$ be an index for which $\frac{a}{d_j} d_{j+1} > c_1$ holds. Then the interval $[d_j, d_{j+1}]$ is such that $[\frac{a}{d_j} d_j, \frac{a}{d_j} d_{j+1}] \equiv [a, e_j]$ with $e_j > c_1$. Hence, by our preceding work we have that $\eta_\alpha[d_j, d_{j+1}] > \eta_\alpha[a, c_1]$ as desired.

From this it follows that $\max_{0 \le i \le \nu-1} \eta_\alpha[c_i, c_{i+1}] < \max_{0 \le i \le \nu-1} \eta_\alpha[d_i, d_{i+1}]$ for each $\underline{d} \in \mathcal{P}$ with $\underline{d} \ne \underline{c}$. To see that this is also true for η_α replaced with η_α^k, k a positive integer, one need only observe that N_α is a strictly

pointwise monotone one-sided operator [9]. What this implies is that $N_\alpha(R)(x) > x^\alpha$ for $x \in [a, b]$ and $R(x) \neq x^\alpha$ and if, $N_\alpha(R_1)(y) > N_\alpha(R_2)(y)$ for some $y \in [a, b]$ then $N_\alpha^k(R_1)(y) > N_\alpha^k(R_2)(y)$ for k a positive integer. From this observation the final result readily follows. ∎

Before applying this theory to some specific examples, we wish to study the problem of finding best starting approximations from $\mathcal{L}_0^1[a, b] \equiv \Pi_1$ for calculating n^{th} roots on [a, b]. In this very simple case, it is possible to give analytical formulas for the best relative approximation to $x^{1/n}$ from Π_1 on [a, b] and, therefore, also for the best starting approximation from Π_1 for calculating n^{th} roots on [a, b]. See reference [13] where this result has also appeared.

Theorem 3. Fix the interval [a, b], $0 < a < b$. Then the best (linear) relative approximation to $x^{1/n}$ on [a, b] from Π_1, $\tilde{p}(x) = \alpha x + \beta$, is given by

$$\alpha = \frac{(b^{1/n} - a^{1/n})(1 - \lambda)}{b - a}, \tag{4}$$

$$\beta = \frac{(ba^{1/n} - ab^{1/n})(1 - \lambda)}{b - a} \tag{5}$$

where

$$\lambda = \left\| \frac{x^{1/n} - \tilde{p}(x)}{x^{1/n}} \right\| = \min_{p \in \Pi_1} \left\| \frac{x^{1/n} - p(x)}{x^{1/n}} \right\| = \frac{w - 1}{w + 1}, \tag{6}$$

and

$$w = \frac{n}{n - 1} \left(\frac{ba^{1/n} - ab^{1/n}}{b - a} \right) \left(\frac{(n - 1)(b^{1/n} - a^{1/n})}{ba^{1/n} - ab^{1/n}} \right)^{1/n}. \tag{7}$$

Proof: By referring to the proof of Theorem 2, we know that the error curve $E_n(x) = 1 - \frac{\tilde{p}(x)}{x^{1/n}} = 1 - \frac{\alpha x + \beta}{x^{1/n}}$ must have precisely three extreme

points, a, ξ, b, $a < \xi < b$ and that $\tilde{p}$ will satisfy the following system (the unknowns are α, β, ξ and λ)

$$1 - a^{-1/n}(\alpha a + \beta) = \lambda$$

$$1 - \xi^{-1/n}(\alpha \xi + \beta) = -\lambda$$

$$1 - b^{-1/n}(\alpha b + \beta) = \lambda$$

$$(n - 1)\alpha\xi^{-1/n} - \beta\xi^{-(n+1)/n} = 0$$

where the fourth equation is the derivative of the error curve at ξ set equal to 0. Also, since the best linear relative approximation is unique, we have that there exists one and only one solution to this system. Solving simultaneously for α and β in terms of $1 - \lambda$ in equations 1 and 3 gives (4) and (5) of the theorem. Substituting this in equation 4 gives an expression for ξ and then substituting all these values in equation 2 gives the formula for λ. ∎

Corollary 1. The best starting approximation from Π_1 for calculating n^{th} roots is $p^*(x) \equiv \gamma_{1/n}\tilde{p}(x)$ where $\tilde{p}$ is defined in Theorem 3 and $\gamma_{1/n}$ is given by (3) with $\alpha = \frac{1}{n}$. ∎

4. EXAMPLES

In this section we give specific examples of the above theory for computing square roots, reciprocal square roots using a divide-free Newton iteration and, finally, cube roots. In the first two examples we shall only use best starting approximations from Π_1 and consider what

happens when at most two Newton iterations are required. For the cube root case we shall also consider other classes of rational functions for the initialization of the Newton iteration.

A. Square Roots

To develop an algorithm for computing square roots based upon the preceding theory, we must first select an interval of application. Any interval of the form [a, 4a], a > 0 will do; however, the reasonable choices are intervals such as $[\frac{1}{8}, \frac{1}{2}]$, $[\frac{1}{4}, 1]$ or $[\frac{1}{2}, 2]$. We shall use $[\frac{1}{2}, 2]$ as our interval of application here. Thus, an algorithm for calculating square roots based on $[\frac{1}{2}, 2]$ will have the following components. First of all, the algorithm will have a scaling feature. Thus, to find $\sqrt{y}$, y > 0 the algorithm will first scale y; that is, it will calculate m, an integer for which $y = 2^{2m}x$ and $x \in (\frac{1}{2}, 2]$. Next, it will compute one or more Newton iterates, $N_{1/2}(h)(x) = \frac{1}{2}(h(x) + \frac{x}{h(x)})$, to calculate $\sqrt{x}$ using the above theory with a best linear polynomial starting approximation on $[\frac{1}{2}, 2]$. It will then multiply (shift) this final value by 2^m and return this for the value $\sqrt{y}$. For $\alpha = \frac{1}{2}$ the formulas of Theorem 3 and (3) for the interval [a, b] reduce to

$$\lambda_{1/2} = -\left[\frac{b^{1/4} - a^{1/4}}{b^{1/4} + a^{1/4}}\right]^2 \tag{8}$$

$$\alpha = \frac{1 - \lambda}{b^{1/2} + a^{1/2}} \tag{9}$$

$$\beta = a^{1/2}b^{1/2}\alpha \tag{10}$$

$$\gamma_{1/2} = (1 - \lambda_{1/2}^2)^{-1/2} . \tag{11}$$

Thus, the best relative approximation to $\sqrt{x}$ on $[\frac{1}{2}, 2]$ is $\tilde{p}(x)$=.4852813742x + .4852813742 with deviation $\lambda_{1/2}$ = -.02943725152 so that the best starting approximation from Π_1 on $[\frac{1}{2}, 2]$ for calculating square roots is $p^*(x)$ = $\gamma_{1/2}\tilde{p}(x)$ = .4848608528(x + 1) with (relative) error $\eta_{1/2}[\frac{1}{2},2]$=.000396107 which can be calculated by evaluating $\left|1 - \frac{N_{1/2}(p^*)(x)}{\sqrt{x}}\right|$ for x either of the endpoints of the interval [a, b] (here x = $\frac{1}{2}$ or 2). Thus, after one Newton iteration, we are guaranteed an approximation for $\sqrt{x}$ with (relative) error $\leq$ .396107 x 10^{-4} on $[\frac{1}{2}, 2]$. In addition, $\eta^2_{1/2}[\frac{1}{2}, 2]$ = 7.841 x 10^{-8}, so that if one were to calculate square roots by this method using two Newton iterates the (relative) accuracy of the approximation on $[\frac{1}{2}, 2]$ to $\sqrt{x}$ would be $\leq$ 7.841 x 10^{-8}. (All calculations were done on a Texas Instruments SR-56).

Next, consider the case where the interval $[\frac{1}{2}, 2]$ is to be subdivided into two subintervals. By Theorem 2, the optimal subdivision is $[\frac{1}{2}, 2]$ = $[\frac{1}{2}, 1] \cup [1, 2]$. Thus, to calculate $\sqrt{y}$, y > 0 in this case, the algorithm would again scale y, $y = 2^{2m}x$, $x \in (\frac{1}{2}, 2]$. However, it must next determine which subinterval contains x by comparing x with 1. After doing this the algorithm proceeds as above with the best starting linear polynomials given below, using the polynomial that corresponds to the interval containing x.

For $[\frac{1}{2}, 1]$ the best starting approximation from Π_1 for calculating square roots on $[\frac{1}{2}, 1]$ is $p^*(x) = \gamma_{1/2}p(x)$ = .5901785321x + .4173192421 with $\eta_{1/2}[\frac{1}{2}, 1]$ = 2.789912 x 10^{-5} and $\eta^2_{1/2}[\frac{1}{2}, 1]$ = 3.8 x 10^{-10}.

For the second subinterval, [1, 2], the best starting approximation from Π_1 for calculating square roots on [1, 2] is $p^*(x)$ = .4173192421x+.5901785321 so that $\eta_{1/2}[1, 2]$ = 2.789912 x 10^{-5} = $\eta_{1/2}[\frac{1}{2}, 1]$ and $\eta^2_{1/2}[1, 2]$ = $\eta^2_{1/2}[\frac{1}{2}, 1]$ = 3.8 x 10^{-10} (as Theorem 2 predicts).

Finally, for the square root case let us consider this algorithm when we subdivide into three subintervals. In this case, we shall scale to the interval $[\frac{1}{4}, 1]$ so that a comment with respect to [3] can be made (the choice of $[\frac{1}{4}, 1]$ rather than $[\frac{1}{2}, 2]$ in that paper was simply a preference for how we viewed the fixed point numbers in our microprocessor). In this case the theoretically optimal subdivision is $[\frac{1}{4},1]=[\frac{1}{4},(\frac{1}{4})^{2/3}]\ [(\frac{1}{4})^{2/3}, (\frac{1}{4})^{1/3}]\ [(\frac{1}{4})^{1/3}, 1]$. Repeating the above calculations for these three intervals, gives $p^*(x)$ $= .8879377727x + .2796828727$ for $[\frac{1}{4}, (\frac{1}{4})^{2/3}]$; $p^*(x) = .7047566772x$ $+ .3523783386$ for $[(\frac{1}{4})^{2/3}, (\frac{1}{4})^{1/3}]$; and $p^*(x) = .5593657454x + .4439688863$ for $[(\frac{1}{4})^{1/3},1]$ with $\eta_{1/2} = 5.54139\text{x}10^{-6}$ for all three subintervals. In all cases, $\eta_{1/2}^2$ is of the order of 1×10^{-11} in this example. Since 2^{-16} is of the order 1.53×10^{-5}, it appears to be necessary to partition $[\frac{1}{4}, 1]$ into three subintervals to obtain an accuracy of at least 2^{-16}. Since the domain of application in [3] was $X = \{\frac{j}{2^{16}}\}_{j=2^{14}+1}^{2^{16}}$, the above theory implies that X should be partitioned into $X_1 \cup X_2 \cup X_3$ where $X_1 = \{\frac{j}{2^{16}}\}_{j=2^{14}+1}^{26,007}$, $X_2=\{\frac{j}{2^{16}}\}_{j=26,008}^{41,284}$ and $X_3 = \{\frac{j}{2^{16}}\}_{j=41,285}^{2^{16}}$. For this partition the errors $\eta_{1/2}[X_1]$, $\eta_{1/2}[X_2]$ and $\eta_{1/2}[X_3]$ will all be less than $\eta_{1/2}$ of above and in this case not equal. However, in [3] the absolute error was the measure of accuracy so that this subdivision was modified. Specifically, for the subdivision $X = X_1 \cup X_2 \cup X_3$ given above we have that ($\eta = \eta_{1/2}$)

$$|\sqrt{x} - N_{1/2}(p^*(\cdot, X_i))(x)| \leq \eta\varepsilon_i\sqrt{x}$$

for $x \in X_i$, $i = 1, 2$ or 3 where $\varepsilon_1 = (\frac{1}{4})^{1/3}$, $\varepsilon_2 = (\frac{1}{4})^{1/6}$ and $\varepsilon_3 = 1$ and $p^*(\cdot, X_i)$ denotes the best linear (relative) starting approximation for calculating square roots on X_i, $i = 1, 2$ or 3. Thus, our final choice for our partition was $X = Y_1 \cup Y_2 \cup Y_3$ where $Y_1 = (\frac{1}{4}, \frac{13}{32}] \cap X$, $Y_2 = (\frac{13}{32}, \frac{21}{32}) \cap X$ and $Y_3 = (\frac{21}{32}, 1] \cap X$. In addition, the polynomials $p^*(x, X_i) = \alpha_i x + \beta_i$

were also modified so that the product $\alpha_i x$ could be computed through the use of two shifts and one add and the required accuracy after one Newton iteration was still 2^{-16} (absolute error). As remarked earlier, it is not known how to optimally subdivide an interval for the corresponding absolute error problem. To some degree, this would involve estimating the error of this procedure as a function of the interval and this is a difficult problem. In fact, even in the classical theory of best uniform approximation not much is known in this regard.

B. Reciprocal Square Roots - Divide-Free Iteration

This amounts to applying the above theory with $\alpha = -\frac{1}{2}$ $(n = -2)$. In this case $N_{-1/2}(h)(x) = \frac{h(x)}{2}[3 - x \cdot h^2(x)]$. In developing an algorithm using this iteration we must again scale numbers as in the square root case. Thus, we shall assume that the scaling will be done with respect to the interval $[\frac{1}{2}, 2]$. In what follows, we shall give the best linear starting approximations for this algorithm when one uses the interval $[\frac{1}{2}, 2]$, subdivides it into two subintervals and subdivides it into six subintervals.

Now for this particular iteration on an interval [a, b], the formulas of theorem 3 and (3) reduce to

$$(12)\qquad \lambda_{-1/2} = \frac{2(b + a^{1/2}b^{1/2} + a)^{3/2} - 3^{3/2}a^{1/2}b^{1/2}(b^{1/2} + a^{1/2})}{2(b + a^{1/2}b^{1/2} + a)^{3/2} + 3^{3/2}a^{1/2}b^{1/2}(b^{1/2} + a^{1/2})}$$

$$(13)\qquad \alpha = -\frac{(1 - \lambda)}{a^{1/2}b^{1/2}(b^{1/2} + a^{1/2})}$$

$$(14)\qquad \beta = -(b + a^{1/2}b^{1/2} + a)\alpha$$

$$(15)\qquad \gamma_{-1/2} = \left(\frac{3}{3 - \lambda^2_{-1/2}}\right)^{1/2}$$

Thus, the best starting approximation from Π_1 for calculating reciprocal square roots on $[\frac{1}{2}, 2]$ via this iteration is $p^*(x) = -.4314166817x + 1.509958386$, with $\eta_{-1/2}[\frac{1}{2}, 2] = 1.048824252 \times 10^{-2}$ and $\eta^2_{-1/2}[\frac{1}{2}, 2] = 1.64427987 \times 10^{-4}$.

For subdividing $[\frac{1}{2}, 2]$ into two subintervals, we have by the previous theory, $[\frac{1}{2}, 2] = [\frac{1}{2}, 1] \cup [1, 2]$ is the optimal partition. For the subinterval $[\frac{1}{2}, 1]$, the best starting approximation from Π_1 for calculating reciprocal square roots on $[\frac{1}{2}, 1]$ via this iteration is $p^*(x) = -.8100537518x + 1.787875129$ with $\eta_{-1/2}[\frac{1}{2}, 1] = 7.37705125 \times 10^{-4}$ and $\eta^2_{-1/2}[\frac{1}{2}, 1] = 8.16115 \times 10^{-7}$. Likewise, for the interval $[1, 2]$, $p^*(x) = -.2863972505x + 1.264218627$ with $\eta_{-1/2}[1, 2] = \eta_{-1/2}[\frac{1}{2}, 1]$ and $\eta^2_{-1/2}[1, 2] = \eta^2_{-1/2}[1, 2]$.

Finally, let us consider the form of this sort of an algorithm where it is desired to subdivide $[\frac{1}{2}, 2]$ into six subintervals. In this case the optimal partition is $[\frac{1}{2}, 2] = [\frac{1}{2}, (\frac{1}{2})^{2/3}] \cup [(\frac{1}{2})^{2/3}, (\frac{1}{2})^{1/3}] \cup [(\frac{1}{2})^{1/3}, 1] \cup [1, 2^{1/3}] \cup [2^{1/3}, 2^{2/3}] \cup [2^{2/3}, 2]$. We shall only consider the interval $[1, 2^{1/3}]$ since the other intervals can be treated in a like manner. Thus, for $[1, 2^{1/3}]$ by direct calculations, $p^*(x) = -.4186992113x + 1.416201135$ with $\eta_{-1/2}[1, 2^{1/3}] = 9.352785 \times 10^{-6}$ and $\eta^2_{-1/2}[1, 2^{1/3}] = 1.35 \times 10^{-10}$. Note that, in addition to the cost of the scaling and determining which subinterval contains the scaled number, this algorithm for one Newton iteration, consists of three Multiplies, one add and one shift.

C. Cube Roots

For this particular example we will also consider some nonlinear best starting approximation. As before, in designing a cube root routine of this sort one must first select an interval of application. The interval we shall use is $[\frac{1}{8}, 1]$. Thus, the final algorithm for computing $\sqrt[3]{y}$, y real, would have a scaling routine which would (i) change the sign of y if $y < 0$ and also change the sign (to negative) of the computed cube root of -y prior to returning a final approximation, and (ii) scale y (assume $y > 0$), i.e. compute $y = 2^{3m}x$ where m is an integer and $x \in (\frac{1}{8}, 1]$. Then, the cube root of x can be calculated by any one of the following routines; the result will be multiplied (shifted) by 2^m and returned for $\sqrt[3]{y}$. In what follows, we shall give the best linear starting approximation for the full interval $[\frac{1}{8}, 1]$ and partitions of this interval into 3 and 6 subintervals. In all cases, we shall also give the (relative) error after one and two Newton iterates. In addition, for the interval $[\frac{1}{8}, 1]$ and the partition of this interval into 3 subintervals, we shall give the best starting approximations from $\mathcal{R}_n^m$, $0 \leq m, n, m + n = k, k = 1, 2, 3$ where m and n for fixed k are chosen so that the best starting approximation from $\mathcal{R}_{\bar{n}}^{\bar{m}}$, $(\bar{m}, \bar{n}) \neq (m, n)$, $0 \leq \bar{m}, \bar{n}, \bar{m} + \bar{n} = m + n$ does not give a better relative approximation for $\sqrt[3]{x}$ after one newton iteration. These best starting approximations were calculated on a CDC-6400 where we found the best relative approximation to $\sqrt[3]{x}$ on the interval in question discretized into equally spaced mesh points with a step size of $h = \frac{1}{256}$ using [7]. We then multiplied this function by the appropriate $\gamma_{1/3}$ given by (3) using the respective $\lambda_{1/3}$.

First, for $[\frac{1}{8}, 1]$. For the class Π_1, we find that the best starting approximation is $p^*(x) = .6055481056x + .4541610792$ with $\eta_{1/3}[\frac{1}{8}, 1] = 3.30112 \times 10^{-3}$ and $\eta^2_{1/3} = 1.08496 \times 10^{-5}$. In addition, the best starting approximation from $\Pi_1 = \mathcal{R}^1_0[\frac{1}{8}, 1]$ is better than the best starting approximation from $\mathcal{R}^0_1$.

Next, for k = 2, we find that for the three classes, $\Pi_2 = \mathcal{R}^2_0[\frac{1}{8}, 1]$, $\mathcal{R}^1_1[\frac{1}{8}, 1]$, $\mathcal{R}^0_2[\frac{1}{8}, 1]$ the best starting approximation from $\mathcal{R}^1_1$ is the best since the best relative approximation to $\sqrt[3]{x}$ from $\mathcal{R}^1_1[\frac{1}{8}, 1]$ on $[\frac{1}{8}, 1]$ gives a relative error of approximately $.650 \times 10^{-2}$; whereas, this corresponding error is $.159 \times 10^{-1}$ and $.418 \times 10^{-1}$ from $\mathcal{R}^2_0[\frac{1}{8}, 1]$ and $\mathcal{R}^0_2[\frac{1}{8}, 1]$, respectively. Using the code [7], we find that the best relative approximation to $\sqrt[3]{x}$ on $[\frac{1}{8}, 1]$ is $\tilde{R}(x) = .1477442896 - \frac{.8414551422}{.7387462419 + x}$ with relative error $\lambda_{1/3} = .6500719120 \times 10^{-2}$ so that $\gamma_{1/3} = (1 - \lambda^2_{1/3})^{-2/3}$ = 1.000028174 and the best starting approximation is $R^*(x) = 1.477484521 - \frac{.8414788493}{.7387462419 + x}$ with $\eta_{1/3}[\frac{1}{8}, 1] = 4.226074 \times 10^{-5}$ and $\eta^2_{1/3}[\frac{1}{8}, 1] = 1.78 \times 10^{-9}$.

For k = 3 by the same procedure we find that $\mathcal{R}^2_1[\frac{1}{8}, 1]$ is the preferable class from $\mathcal{R}^3_0[\frac{1}{8}, 1]$, $\mathcal{R}^2_1[\frac{1}{8}, 1]$, $\mathcal{R}^1_2[\frac{1}{8}, 1]$ and $\mathcal{R}^0_3[\frac{1}{8}, 1]$. For this class, the best starting approximation is $R^*(x) = .2437995493x + .8898929185 - \frac{.1698975861}{.2796064148+x}$ with $\eta_{1/3}[\frac{1}{8}, 1] = 8.438 \times 10^{-7}$ and $\eta^2_{1/3}[\frac{1}{8}, 1] = 0.0$ (on our SR-56, should be of the order 10^{-14}).

Next, for the (optimal) partition $[\frac{1}{8},1] = [\frac{1}{8},\frac{1}{4}] \cup [\frac{1}{4},\frac{1}{2}] \cup [\frac{1}{2},1]$. Here we shall only give the results for $[\frac{1}{2}, 1]$. The best linear starting approximation for calculating cube roots on $[\frac{1}{2}, 1]$ is $p^*(x) = .4153501945x + .5913005214$ with $\eta_{1/3}[\frac{1}{2}, 1] = 4.407136 \times 10^{-5}$ and $\eta^2_{1/3}[\frac{1}{2}, 1] = 1.94 \times 10^{-9}$. As before on $[\frac{1}{8}, 1]$, this class is preferrable to $\mathcal{R}^0_1[\frac{1}{2}, 1]$.

Once again, $\mathcal{R}^1_1[\frac{1}{2}, 1]$ is the preferrable class to use from the set $\mathcal{R}^2_0[\frac{1}{2}, 1]$, $\mathcal{R}^1_1[\frac{1}{2}, 1]$ and $\mathcal{R}^0_2[\frac{1}{2}, 1]$. The best starting approximation from $\mathcal{R}^1_1[\frac{1}{2}, 1]$ for calculating cube roots on $[\frac{1}{2}, 1]$ is $R^*(x) = 1.790709274 - \frac{1.347475985}{1 + .7035864292x}$ with $\eta_{1/3}[\frac{1}{2}, 1] = 6.503 \times 10^{-8}$ and $\eta^2_{1/3}[\frac{1}{2}, 1] = 0.0$ (on our SR-56, should be of the order of 10^{-16}).

From the classes $\mathcal{L}_0^3[\frac{1}{2}, 1]$, $\mathcal{L}_1^2[\frac{1}{2}, 1]$, $\mathcal{L}_2^1[\frac{1}{2}, 1]$ and $\mathcal{L}_3^0[\frac{1}{2}, 1]$, we again find that the class $\mathcal{L}_1^2[\frac{1}{2}, 1]$ will give the most accurate (relative) approximation. The best starting approximation from $\mathcal{L}_1^2[\frac{1}{2}, 1]$ is $R^*(x) = .1662848358 + 1.096040958x - \frac{.4105032829}{.5649335816 + x}$ with $\eta_{1/3}[\frac{1}{2}, 1] = 1.5 \times 10^{-10}$ and $\eta_{1/3}^2[\frac{1}{2}, 1]$ of the order 10^{-20}.

Finally, if one partitions $[\frac{1}{8}, 1]$ into 6 subintervals the optimal partition is $[\frac{1}{8},1]=[\frac{1}{8},(\frac{1}{2})^{5/2}]\cup[(\frac{1}{2})^{5/2},\frac{1}{4}]\cup[\frac{1}{4},\frac{\sqrt{2}}{4}]\cup[\frac{\sqrt{2}}{4},\frac{1}{2}]\cup[\frac{1}{2},\frac{\sqrt{2}}{2}]\cup[\frac{\sqrt{2}}{2},1]$. For the interval $[\frac{\sqrt{2}}{2}, 1]$ the best linear starting approximation is $p^*(x) = .3731163725x + .6285515591$ with $\eta_{1/3}[\frac{\sqrt{2}}{2}, 1] = 2.77582 \times 10^{-6}$ and $\eta_{1/3}^2[\frac{\sqrt{2}}{2}, 1]$ of the order 10^{-12}.

The above variety of possibilities indicates the flexibility of these sort of algorithms. Thus, in designing a specific subroutine for calculating a fixed n^{th} root one can try to minimize the effect of such things as machines that are relatively slow in computing a divide and so on.

REFERENCES

1. N. I. Achiesen, "Theorey of Approximations", Ungar Publishing Co., New York, 1956.

2. M. Andrews, S. F. McCormick and G. D. Taylor, Evaluation of the square root function on microprocessors, to appear.

3. M. Andrews, D. Eggerding, S. F. McCormick and G. D. Taylor, Square root routines for microprocessors, to appear.

4. T. C. Chen, The automatic computation of exponentials, logarithms, ratios and square roots, IBM Research Memo, RJ970 (#16884), February, 1972, San Jose, California.

5. W. J. Cody, Double-precision square root for the CDC-3600, Comm. ACM, 7 (1964), 715-718.

6. J. B. Gibson, Optimal rational starting approximations, J. of Approx. Theory 12 (1974), 182-189.

7. E. H. Kaufman, Jr. and G. D. Taylor, Uniform rational approximation of functions of several variables, Int. J. For Numer. Meth. in Engin., 9 (1975), 297-323.

8. R. F. King and D. L. Phillips, The logarithmic error and Newton's method for the square root, Comm. ACM, 12 (1969), 87-88.

9. G. Meinardus and G. D. Taylor, Optimal starting approximations for iterative schemes,.J. of Approx. Theory, 9 (1970), 1-19.

10. D. G. Moursund, Optimal starting values for the Newton-Raphson Calculation of x, Comm. ACM, 10 (1967), 430-432.

11. D. G. Moursund and G. D. Taylor, Optimal starting values for the Newton-Raphson calculation of inverses of certain functions, SIAM J. Numer. Anal., 5 (1968), 138-150.

12. I. Ninomiya, Best rational starting approximations and improved Newton iteration for square root, Math. Comp. 24 (1970), 391-404.

13. D. L. Phillips, Generalized logarithmic error and Newton's method for the m^{th} root, Math. Comp. 24 (1970), 383-389.

14. P. H. Sterbenz and C. T. Fike, Optimal starting approximations for Newton's method, Math. Comp. 23 (1969), 313-318.

15. G. D. Taylor, Optimal starting approximations for Newton's method, J. Approx. Theory 3 (1970), 156-163.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(18) (19) REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: AFOSR-TR-77-1261
2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle): (16) OPTIMAL PARTITIONING OF NEWTON'S METHOD FOR CALCULATING ROOTS.
5. TYPE OF REPORT & PERIOD COVERED: (9) Interim rept.,
6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s): (10) Gunter Meinardus and G. D. Taylor
8. CONTRACT OR GRANT NUMBER(s): (15) AFOSR-76-2378

9. PERFORMING ORGANIZATION NAME AND ADDRESS:
Colorado State University
Department of Mathematics
Fort Collins, CO 80523
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 61102F (16) 2304 (17) A3

11. CONTROLLING OFFICE NAME AND ADDRESS:
AFOSR/NM
Bldg. 410
Bolling AFB, D.C. 20332
12. REPORT DATE: (11) May 1977
13. NUMBER OF PAGES: 21

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office): (12) 22p.
15. SECURITY CLASS. (of this report): UNCLASSIFIED
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release, distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Newton's method for calculating roots

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

In this paper an algorithm for calculating roots is given that is Newton's method initialized with a piecewise best starting approximation. The piecewise best starting approximation corresponds to a partition of the interval of the domain of Newton's method and it is shown how to choose this partition to be optimal. Explicit formulas are given when linear polynomials are used for the best starting approximations. Specific examples are given for square roots, cube roots and reciprocal square roots.

LB

16. J. S. Walther, A unified algorithm for elementary functions, Proc. AFIPS, 1971, Spring Joint Computer Conference, pp. 379-385.

17. M. W. Wilson, Optimal starting approximations for generating square root for slow or no divide, Comm. ACM, 13 (1970).